

Error control coding: An introduction to linear block codes

Adrish Banerjee

Department of Electrical Engineering
Indian Institute of Technology Kanpur
Kanpur, Uttar Pradesh
India

Mar. 21, 2016



Lecture #4: Decoding of linear block codes



Decoding of linear block codes

Basic idea:

- The received vector $\mathbf{r}$ has 2^n possibilities, regardless of what codeword is transmitted.



Decoding of linear block codes

Basic idea:

- The received vector $\mathbf{r}$ has 2^n possibilities, regardless of what codeword is transmitted.
- Any decoding scheme used at the receiver is a rule to partition the 2^n possible received vectors into 2^k disjoint subsets $D_1, D_2, \dots, D_{2^k}$, such that the codeword $\mathbf{v}_i$ is contained in the subset D_i for $1 \leq i \leq 2^k$.



Decoding of linear block codes

Basic idea:

- The received vector $\mathbf{r}$ has 2^n possibilities, regardless of what codeword is transmitted.
- Any decoding scheme used at the receiver is a rule to partition the 2^n possible received vectors into 2^k disjoint subsets $D_1, D_2, \dots, D_{2^k}$, such that the codeword $\mathbf{v}_i$ is contained in the subset D_i for $1 \leq i \leq 2^k$.
- The partition is based on linear structure of the code.



Decoding of linear block codes

- Let $\mathbf{v}_1 = \mathbf{0}, \mathbf{v}_2, \dots, \mathbf{v}_{2^k}$ be the 2^k codewords in an (n, k) linear block code.



Decoding of linear block codes

- Let $\mathbf{v}_1 = \mathbf{0}, \mathbf{v}_2, \dots, \mathbf{v}_{2^k}$ be the 2^k codewords in an (n, k) linear block code.
- Form an array of vectors from vector space, V_n as follows:



Decoding of linear block codes

- Let $\mathbf{v}_1 = \mathbf{0}, \mathbf{v}_2, \dots, \mathbf{v}_{2^k}$ be the 2^k codewords in an (n, k) linear block code.
- Form an array of vectors from vector space, V_n as follows:
 - (i) Arrange the 2^k codewords as the top row of the array with $\mathbf{v}_1 = \mathbf{0}$ as the first element.



Decoding of linear block codes

- Let $\mathbf{v}_1 = \mathbf{0}, \mathbf{v}_2, \dots, \mathbf{v}_{2^k}$ be the 2^k codewords in an (n, k) linear block code.
- Form an array of vectors from vector space, V_n as follows:
 - (i) Arrange the 2^k codewords as the top row of the array with $\mathbf{v}_1 = \mathbf{0}$ as the first element.
 - (ii) Suppose $j - 1$ rows of the array have been formed. Choose a vector $\mathbf{e}_j$ from V_n which is not in the previous $j - 1$ rows.



Decoding of linear block codes

- Let $\mathbf{v}_1 = \mathbf{0}, \mathbf{v}_2, \dots, \mathbf{v}_{2^k}$ be the 2^k codewords in an (n, k) linear block code.
- Form an array of vectors from vector space, V_n as follows:
 - (i) Arrange the 2^k codewords as the top row of the array with $\mathbf{v}_1 = \mathbf{0}$ as the first element.
 - (ii) Suppose $j - 1$ rows of the array have been formed. Choose a vector $\mathbf{e}_j$ from V_n which is not in the previous $j - 1$ rows.
 - (iii) Form the j^{th} row by adding $\mathbf{e}_j$ to each codeword $\mathbf{v}_i$ in the top row and placing $\mathbf{e}_j + \mathbf{v}_i$ under $\mathbf{v}_i$.



Decoding of linear block codes

- Let $\mathbf{v}_1 = \mathbf{0}, \mathbf{v}_2, \dots, \mathbf{v}_{2^k}$ be the 2^k codewords in an (n, k) linear block code.
- Form an array of vectors from vector space, V_n as follows:
 - (i) Arrange the 2^k codewords as the top row of the array with $\mathbf{v}_1 = \mathbf{0}$ as the first element.
 - (ii) Suppose $j - 1$ rows of the array have been formed. Choose a vector $\mathbf{e}_j$ from V_n which is not in the previous $j - 1$ rows.
 - (iii) Form the j^{th} row by adding $\mathbf{e}_j$ to each codeword $\mathbf{v}_i$ in the top row and placing $\mathbf{e}_j + \mathbf{v}_i$ under $\mathbf{v}_i$.
 - (iv) Continue until all the vectors from V_n appear in the array.



Decoding of linear block codes

- Let $\mathbf{v}_1 = \mathbf{0}, \mathbf{v}_2, \dots, \mathbf{v}_{2^k}$ be the 2^k codewords in an (n, k) linear block code.
- Form an array of vectors from vector space, V_n as follows:
 - (i) Arrange the 2^k codewords as the top row of the array with $\mathbf{v}_1 = \mathbf{0}$ as the first element.
 - (ii) Suppose $j - 1$ rows of the array have been formed. Choose a vector $\mathbf{e}_j$ from V_n which is not in the previous $j - 1$ rows.
 - (iii) Form the j^{th} row by adding $\mathbf{e}_j$ to each codeword $\mathbf{v}_i$ in the top row and placing $\mathbf{e}_j + \mathbf{v}_i$ under $\mathbf{v}_i$.
 - (iv) Continue until all the vectors from V_n appear in the array.
- The array is called a *standard array*.



Decoding of linear block codes

$\mathbf{v}_1 = \mathbf{0}$	$\mathbf{v}_2$	$\cdots$	$\mathbf{v}_i$	$\cdots$	$\mathbf{v}_{2^k}$
$\mathbf{e}_2$	$\mathbf{e}_2 + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_{2^k}$
$\mathbf{e}_3$	$\mathbf{e}_3 + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_3 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_3 + \mathbf{v}_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$\mathbf{e}_{2^{n-k}}$	$\mathbf{e}_{2^{n-k}} + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_{2^{n-k}} + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_{2^{n-k}} + \mathbf{v}_{2^k}$

Standard array

Decoding of linear block codes

For a (6,3) linear code generated by the following matrix,

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \mathbf{g}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

standard array is shown in next two slides.

Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

Navigation icons: back, forward, search, etc.

Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_5$	$\mathbf{v}_6$	$\mathbf{v}_7$	$\mathbf{v}_8$
(0 0 0 0 0 0)	(1 1 0 1 1 0)	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
(0 1 0 0 0 0)	(1 0 0 1 1 0)	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)

Columns $\mathbf{v}_2 - \mathbf{v}_4$ are listed in the previous page.

Navigation icons: back, forward, search, etc.

Decoding of linear block codes

- Every vector in V_n appears exactly once in the standard array.



Decoding of linear block codes

- Every vector in V_n appears exactly once in the standard array.
 - This follows from the construction rule of the standard array. Proof by contradiction.



Decoding of linear block codes

- Every vector in V_n appears exactly once in the standard array.
 - This follows from the construction rule of the standard array. Proof by contradiction.
- No two vectors in the same row of a standard array are identical

Decoding of linear block codes

- Every vector in V_n appears exactly once in the standard array.
 - This follows from the construction rule of the standard array. Proof by contradiction.
- No two vectors in the same row of a standard array are identical
 - This follows from the fact that all code vectors of C are distinct.

Decoding of linear block codes

- Every vector in V_n appears exactly once in the standard array.
 - This follows from the construction rule of the standard array. Proof by contradiction.
- No two vectors in the same row of a standard array are identical
 - This follows from the fact that all code vectors of C are distinct.
- Each row is called a *coset*.



Decoding of linear block codes

- Every vector in V_n appears exactly once in the standard array.
 - This follows from the construction rule of the standard array. Proof by contradiction.
- No two vectors in the same row of a standard array are identical
 - This follows from the fact that all code vectors of C are distinct.
- Each row is called a *coset*.
- There are exactly 2^{n-k} cosets.



Decoding of linear block codes

- Every vector in V_n appears exactly once in the standard array.
 - This follows from the construction rule of the standard array. Proof by contradiction.
- No two vectors in the same row of a standard array are identical
 - This follows from the fact that all code vectors of C are distinct.
- Each row is called a *coset*.
- There are exactly 2^{n-k} cosets.
- The first element of each coset is called the *coset leader*. (Any element in a coset can be used as its coset leader. This does not change the elements of the coset, it changes the order of them.)

Decoding of linear block codes

- All 2^k elements of a coset have the same syndrome as their coset leader, since

$$\mathbf{s} = (\mathbf{e}_j + \mathbf{v}_i)\mathbf{H}^T = \mathbf{e}_j\mathbf{H}^T + \mathbf{v}_i\mathbf{H}^T = \mathbf{e}_j\mathbf{H}^T$$

Decoding of linear block codes

- All 2^k elements of a coset have the same syndrome as their coset leader, since

$$\mathbf{s} = (\mathbf{e}_j + \mathbf{v}_i)\mathbf{H}^T = \mathbf{e}_j\mathbf{H}^T + \mathbf{v}_i\mathbf{H}^T = \mathbf{e}_j\mathbf{H}^T$$

- The 2^k elements of a coset are the 2^k solutions to the syndrome equations.



Decoding of linear block codes

- All 2^k elements of a coset have the same syndrome as their coset leader, since

$$\mathbf{s} = (\mathbf{e}_j + \mathbf{v}_i)\mathbf{H}^T = \mathbf{e}_j\mathbf{H}^T + \mathbf{v}_i\mathbf{H}^T = \mathbf{e}_j\mathbf{H}^T$$

- The 2^k elements of a coset are the 2^k solutions to the syndrome equations.
- Each of the 2^{n-k} coset leaders has a different syndrome. Hence, there is one-to-one correspondence between a coset leader and a syndrome.



Decoding of linear block codes

- The j^{th} column of a standard array.

$$D_j = \{\mathbf{v}_j, \mathbf{e}_2 + \mathbf{v}_j, \mathbf{e}_3 + \mathbf{v}_j, \dots, \mathbf{e}_{2^{n-k}} + \mathbf{v}_j\}$$

contains exactly one codeword.

Decoding of linear block codes

- The j^{th} column of a standard array.

$$D_j = \{\mathbf{v}_j, \mathbf{e}_2 + \mathbf{v}_j, \mathbf{e}_3 + \mathbf{v}_j, \dots, \mathbf{e}_{2^{n-k}} + \mathbf{v}_j\}$$

contains exactly one codeword.

- If $\mathbf{r}$ belongs to column D_j , then $\mathbf{r}$ is decoded into codeword $\mathbf{v}_j$.

Decoding of linear block codes

- The j^{th} column of a standard array.

$$D_j = \{\mathbf{v}_j, \mathbf{e}_2 + \mathbf{v}_j, \mathbf{e}_3 + \mathbf{v}_j, \dots, \mathbf{e}_{2^{n-k}} + \mathbf{v}_j\}$$

contains exactly one codeword.

- If $\mathbf{r}$ belongs to column D_j , then $\mathbf{r}$ is decoded into codeword $\mathbf{v}_j$.
- If $\mathbf{v}_j$ is the transmitted codeword, and the error pattern is a coset leader $\mathbf{e}_i$, then $\mathbf{r} = \mathbf{v}_j + \mathbf{e}_i$ is in column of D_j , which contains $\mathbf{v}_j$ (Correct decoding).



Decoding of linear block codes

- If the error pattern is not a coset leader, then $\mathbf{r}$ is not in column D_j . (incorrect decoding)



Decoding of linear block codes

- If the error pattern is not a coset leader, then $\mathbf{r}$ is not in column D_j . (incorrect decoding)
- Let's say the error pattern $\mathbf{x}$ caused by the channel is in l^{th} coset and under the code vector $\mathbf{v}_i \neq 0$. Then $\mathbf{x} = \mathbf{e}_l + \mathbf{v}_i$ and the received vector is

$$\mathbf{r} = \mathbf{v}_j + \mathbf{x} = \mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_j = \mathbf{e}_l + \mathbf{v}_s.$$

The received vector is in D_s , and decoded as $\mathbf{v}_s$, which is not the transmitted code vector $\mathbf{v}_j$.



Decoding of linear block codes

- If the error pattern is not a coset leader, then $\mathbf{r}$ is not in column D_j . (incorrect decoding)
- Let's say the error pattern $\mathbf{x}$ caused by the channel is in l^{th} coset and under the code vector $\mathbf{v}_i \neq 0$. Then $\mathbf{x} = \mathbf{e}_l + \mathbf{v}_i$ and the received vector is

$$\mathbf{r} = \mathbf{v}_j + \mathbf{x} = \mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_j = \mathbf{e}_l + \mathbf{v}_s.$$

The received vector is in D_s , and decoded as $\mathbf{v}_s$, which is not the transmitted code vector $\mathbf{v}_j$.

- Therefore, decoding is correct if and only if the error pattern is a coset leader, and the 2^{n-k} coset leaders are all the *correctable error patterns*.



Decoding of linear block codes

- To minimize the probability of error, the error patterns most likely to happen should be chosen as coset leaders.



Decoding of linear block codes

- To minimize the probability of error, the error patterns most likely to happen should be chosen as coset leaders.
- For BSC, an error pattern of smaller weight is more probable than an error pattern of higher weight.



Decoding of linear block codes

- To minimize the probability of error, the error patterns most likely to happen should be chosen as coset leaders.
- For BSC, an error pattern of smaller weight is more probable than an error pattern of higher weight.
- Each coset leader should be chosen to be a vector of least weight from the available vectors.



Decoding of linear block codes

- To minimize the probability of error, the error patterns most likely to happen should be chosen as coset leaders.
- For BSC, an error pattern of smaller weight is more probable than an error pattern of higher weight.
- Each coset leader should be chosen to be a vector of least weight from the available vectors.
- This way coset leader has minimum weight in its coset.



Decoding of linear block codes

- To minimize the probability of error, the error patterns most likely to happen should be chosen as coset leaders.
- For BSC, an error pattern of smaller weight is more probable than an error pattern of higher weight.
- Each coset leader should be chosen to be a vector of least weight from the available vectors.
- This way coset leader has minimum weight in its coset.
- The decoding based on standard array is minimum distance decoding (i.e. ML decoding).



Decoding of linear block codes

- Assume that the received vector $\mathbf{r}$ is found in the i^{th} column, and j^{th} coset of the standard array. Then $\mathbf{r}$ is decoded as code vector $\mathbf{v}_j$.



Decoding of linear block codes

- Assume that the received vector $\mathbf{r}$ is found in the i^{th} column, and j^{th} coset of the standard array. Then $\mathbf{r}$ is decoded as code vector $\mathbf{v}_j$.
- Since $\mathbf{r} = \mathbf{e}_i + \mathbf{v}_j$, distance between $\mathbf{r}$ and $\mathbf{v}_j$ is

$$d(\mathbf{r}, \mathbf{v}_j) = w(\mathbf{r} + \mathbf{v}_j) = w(\mathbf{e}_i + \mathbf{v}_j + \mathbf{v}_j) = w(\mathbf{e}_i)$$



Decoding of linear block codes

- Assume that the received vector $\mathbf{r}$ is found in the i^{th} column, and j^{th} coset of the standard array. Then $\mathbf{r}$ is decoded as code vector $\mathbf{v}_j$.
- Since $\mathbf{r} = \mathbf{e}_i + \mathbf{v}_j$, distance between $\mathbf{r}$ and $\mathbf{v}_j$ is

$$d(\mathbf{r}, \mathbf{v}_j) = w(\mathbf{r} + \mathbf{v}_j) = w(\mathbf{e}_i + \mathbf{v}_j + \mathbf{v}_j) = w(\mathbf{e}_i)$$

- Now consider the distance between $\mathbf{r}$ and any other code vector, say $\mathbf{v}_s$,

$$d(\mathbf{r}, \mathbf{v}_s) = w(\mathbf{r} + \mathbf{v}_s) = w(\mathbf{e}_i + \mathbf{v}_j + \mathbf{v}_s) = w(\mathbf{e}_i + \mathbf{v}_s)$$

where $\mathbf{v}_s = \mathbf{v}_i + \mathbf{v}_j$



Decoding of linear block codes

- Since $\mathbf{e}_l$, and $\mathbf{e}_l + \mathbf{v}_s$ are in the same coset, and since $w(\mathbf{e}_l) \leq w(\mathbf{e}_l + \mathbf{v}_s)$, it follows that

$$d(\mathbf{r}, \mathbf{v}_i) \leq d(\mathbf{r}, \mathbf{v}_j)$$



Decoding of linear block codes

- Since $\mathbf{e}_l$, and $\mathbf{e}_l + \mathbf{v}_s$ are in the same coset, and since $w(\mathbf{e}_l) \leq w(\mathbf{e}_l + \mathbf{v}_s)$, it follows that

$$d(\mathbf{r}, \mathbf{v}_i) \leq d(\mathbf{r}, \mathbf{v}_j)$$

- Hence if coset leader is chosen to have minimum weight in its coset, the decoding based on the standard array is the ML decoder.



Decoding of linear block codes

Summary:

Step 1 : Compute the syndrome $\mathbf{s} = \mathbf{rH}^T$.

Step 2 : Find the coset leader $\hat{\mathbf{e}}$ whose syndrome is equal to $\mathbf{s}$.

Step 3 : Decode $\mathbf{r}$ into the estimated codeword

$$\hat{\mathbf{v}} = \mathbf{r} + \hat{\mathbf{e}}$$



Decoding of linear block codes

- Syndrome decoding can be implemented using a look-up table that consists of 2^{n-k} correctable error patterns (coset leaders) and their corresponding syndromes.

$\mathbf{s}_1 = \mathbf{0}$	$\rightarrow$	$\mathbf{e}_1 = \mathbf{0}$
$\mathbf{s}_2$	$\rightarrow$	$\mathbf{e}_2$
$\vdots$	$\vdots$	$\vdots$
$\mathbf{s}_{2^{n-k}}$	$\rightarrow$	$\mathbf{e}_{2^{n-k}}$



Decoding of linear block codes

- Syndrome decoding can be implemented using a look-up table that consists of 2^{n-k} correctable error patterns (coset leaders) and their corresponding syndromes.

$$\begin{array}{ccc} \mathbf{s}_1 = \mathbf{0} & \rightarrow & \mathbf{e}_1 = \mathbf{0} \\ \hline \mathbf{s}_2 & \rightarrow & \mathbf{e}_2 \\ \vdots & \vdots & \vdots \\ \mathbf{s}_{2^{n-k}} & \rightarrow & \mathbf{e}_{2^{n-k}} \end{array}$$

- Syndrome decoding can also be used to perform a combination of error correction and error detection.



Decoding of linear block codes

- Syndrome decoding can be implemented using a look-up table that consists of 2^{n-k} correctable error patterns (coset leaders) and their corresponding syndromes.

$$\begin{array}{ccc} \mathbf{s}_1 = \mathbf{0} & \rightarrow & \mathbf{e}_1 = \mathbf{0} \\ \hline \mathbf{s}_2 & \rightarrow & \mathbf{e}_2 \\ \vdots & \vdots & \vdots \\ \mathbf{s}_{2^{n-k}} & \rightarrow & \mathbf{e}_{2^{n-k}} \end{array}$$

- Syndrome decoding can also be used to perform a combination of error correction and error detection.
- Coset leaders corresponding to the lowest weight error patterns are used for error correction. These are the most likely error patterns.



Decoding of linear block codes

- Syndrome decoding can be implemented using a look-up table that consists of 2^{n-k} correctable error patterns (coset leaders) and their corresponding syndromes.

$$\begin{array}{ccc} \mathbf{s}_1 = \mathbf{0} & \rightarrow & \mathbf{e}_1 = \mathbf{0} \\ \mathbf{s}_2 & \rightarrow & \mathbf{e}_2 \\ \vdots & & \vdots \\ \mathbf{s}_{2^{n-k}} & \rightarrow & \mathbf{e}_{2^{n-k}} \end{array}$$

- Syndrome decoding can also be used to perform a combination of error correction and error detection.
- Coset leaders corresponding to the lowest weight error patterns are used for error correction. These are the most likely error patterns.
- Syndrome corresponding to higher weight (less likely) error patterns are used to declare a *detected error* rather than for correction.



Decoding of linear block codes

Example 3.2: Consider a (6, 3) linear systematic code generated by

$$\mathbf{G} = \left[\begin{array}{ccc|ccc} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{array} \right] = [\mathbf{P} \quad \mathbf{I}]$$

Its parity-check matrix is

$$\mathbf{H} = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{array} \right] = [\mathbf{I}_3 \quad \mathbf{P}^T]$$



Decoding of linear block codes

Encoding:

$$(u_0, u_1, u_2) \longleftrightarrow (v_0, v_1, v_2, u_0, u_1, u_2)$$

where

$$v_0 = u_1 + u_2$$

$$v_1 = u_0 + u_2$$

$$v_2 = u_0 + u_1$$



Decoding of linear block codes

Syndrome look-up table

Syndromes (s_0, s_1, s_2)	Correctable Error Patterns ($e_0, e_1, e_2, e_3, e_4, e_5$)
(0 0 0)	(0 0 0 0 0 0)
(1 0 0)	(1 0 0 0 0 0)
(0 1 0)	(0 1 0 0 0 0)
(0 0 1)	(0 0 1 0 0 0)
(0 1 1)	(0 0 0 1 0 0)
(1 0 1)	(0 0 0 0 1 0)
(1 1 0)	(0 0 0 0 0 1)
(1 1 1)	(1 0 0 1 0 0)



Decoding of linear block codes

Coset Leader, v_1	v_2	v_3	v_4
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

- Correctable error patterns: 7.

◀ ◻ ▶ ◀ ◻ ▶ ◀ ◻ ▶ ◀ ◻ ▶ ◀ ◻ ▶ ◀ ◻ ▶ ◀ ◻ ▶ ◀ ◻ ▶

Decoding of linear block codes

Coset Leader, v_1	v_2	v_3	v_4
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

- Correctable error patterns: 7.
- Detectable error patterns: 8

◀ ◻ ▶ ◀ ◻ ▶ ◀ ◻ ▶ ◀ ◻ ▶ ◀ ◻ ▶ ◀ ◻ ▶ ◀ ◻ ▶ ◀ ◻ ▶

Decoding of linear block codes

Coset Leader, v_1	v_2	v_3	v_4
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

- Correctable error patterns: 7.
- Detectable error patterns: 8
- Undetected decoding errors: 49

Decoding of linear block codes

Coset Leader, v_1	v_5	v_6	v_7	v_8
(0 0 0 0 0)	(1 1 0 1 1 0)	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
(0 1 0 0 0 0)	(1 0 0 1 1 0)	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)